

# AGOL Secure Proxy

## Client Integration Pack

Vector Tile Overlay + Secure Attributes (EA Frame and Buildings)

Release	v1.0
Services	ea_frame, buildings
Capabilities	Vector tile overlay (MVT) + Identify + Query
Auth	Client Key (CK....)

This document provides ready-to-implement endpoints, field whitelists, and code examples for Windows and mobile integration.

## 1. Quick Start

You will receive a Proxy Base URL and a Client Key. Use the style endpoint to display secure overlays, and use identify/query to fetch approved attributes only.

Proxy Base URL	https://YOUR_PROXY_BASE
Client Key	CKXXXXXXXXXXXXXXXXXXXXXX

**Service aliases:** ea\_frame, buildings

```
# List services available to your key
GET /v1/services?key=CK_...

# EA overlay style (MapLibre/Mapbox style JSON)
GET /tiles/ea_frame/style.json?key=CK_...

# Buildings overlay style
GET /tiles/buildings/style.json?key=CK_...

# Identify (tap)
GET /v1/ea_frame/identify?lat=9.0765&lon=7.3986&max_results=3&key=CK_...

# Query (filter/table)
GET /v1/ea_frame/query?where=STATE_NAME%3D%27Kwara%27&outFields=NAT_EA_SN,EA_NAME&resultRecordCount=1
```

## 2. Allowed Fields (Whitelists)

Only the following fields are returned from identify/query responses.

ea_frame	Fields (comma-separated)
	OBJECTID, NAT_EA_SN, STATE_NAME, LGA_NAME, WARD_NAME, EA_NAME, EA_ID, URBAN_RURAL, EST_POP
buildings	Fields (comma-separated)
	OBJECTID, BLD_ID, NAT_EA_SN, STATE_NAME, LGA_NAME, WARD_NAME, EA_ID, BLD_USE, RES_STATUS

### 3. Endpoints

All endpoints are relative to the Proxy Base URL.

Purpose	Method + Path	Notes
List allowed services	GET /v1/services?key=CK_...	Returns aliases enabled for this key.
Overlay style	GET /tiles/{alias}/style.json?key=CK_...	Best for MapLibre/Mapbox-style SDKs.
Vector tile PBF	GET /tiles/{alias}/tile/{z}/{y}/{x}.pbf?key=CK_...	MVT (application/x-protobuf).
Identify (tap)	GET /v1/{alias}/identify?lat=...&lon=...&max_result=...&key=CK_...	Returns key#0 listed attributes only.
Query (filter)	GET /v1/{alias}/query?where=...&outFields=...&resultOffset=...&resultSearchOffset=...&key=CK_...	Supports resultOffset, resultSearchOffset, and key#0.

Note: Keys are required. 401 means missing/invalid/disabled. 403 means key is valid but not allowed for that alias.

## 4. Windows Integration

Recommended: use MapLibre GL JS in a browser (Chrome/Edge) on Windows.

### 4.1 MapLibre GL JS (EA Frame overlay + Identify on click)

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <link href="https://unpkg.com/maplibre-gl@4.7.1/dist/maplibre-gl.css" rel="stylesheet"/>
  <script src="https://unpkg.com/maplibre-gl@4.7.1/dist/maplibre-gl.js"></script>
  <style>#map{position:fixed;inset:0}</style>
</head>
<body>
<div id="map"></div>
<script>
  const PROXY = "https://YOUR_PROXY_BASE";
  const KEY = "CK_...";
  const ALIAS = "ea_frame"; // or "buildings"

  const styleUrl = `${PROXY}/tiles/${ALIAS}/style.json?key=${encodeURIComponent(KEY)}`;

  const map = new maplibregl.Map({
    container: "map",
    style: styleUrl,
    center: [7.3986, 9.0765],
    zoom: 6
  });

  map.on("click", async (e) => {
    const url = `${PROXY}/v1/${ALIAS}/identify?lat=${e.lngLat.lat}&lon=${e.lngLat.lng}&max_results=3`;
    const data = await fetch(url).then(r => r.json());
    console.log("Identify:", data);
  });
</script>
</body>
</html>
```

## 5. Mobile Integration

Most mobile SDKs can load the style endpoint directly. Use identify to fetch attributes on tap.

### 5.1 Android (MapLibre Native) - key steps

- 1) Load style URL:  
`https://YOUR_PROXY_BASE/tiles/ea_frame/style.json?key=CK_...`
- 2) On map tap, call identify:  
`https://YOUR_PROXY_BASE/v1/ea_frame/identify?lat={lat}&lon={lon}&max_results=3&key=CK_...`
- 3) Render attributes from the response.

### 5.2 iOS (MapLibre Native) - key steps

- 1) Load style URL:  
`https://YOUR_PROXY_BASE/tiles/ea_frame/style.json?key=CK_...`
- 2) On tap, call identify using URLSession:  
`https://YOUR_PROXY_BASE/v1/ea_frame/identify?lat={lat}&lon={lon}&max_results=3&key=CK_...`

### 5.3 Flutter and React Native - key steps

Flutter:

- Use a MapLibre/Mapbox-style plugin that supports loading a style URL.
- Call identify/query using the http or dio package.

React Native:

- Use a MapLibre compatible library (or a WebView map).
- Call identify/query using fetch/axios.

Security note: client keys embedded in mobile apps can be extracted. Use per-client restrictions (where\_lock) and rotate keys when needed.